

Twitter Thread by foone



foone
@Foone



love to see a youtube video that references a script that's linked in the comments and it's an obfuscated link to a one-click hoster.

THAT'S DEFINITELY STILL GONNA WORK IN FOUR YEARS

it does still work, but that's because it's only been 2 years.

also, wow, this is horrible.

the script is to generate random kanji/han characters and insert them into minecraft books.

but the video references there being two hotkeys to generate two distinct books, which seemed odd. so I thought I'd check.

it turns out by "random" they mean "pre-randomized and is hardcoded into the 45 kilobyte autohotkey scripot"

I mean, I get it. I don't want to write AutoHotKey scripts either. it's a horrible, horrible language.

BUT STILL

it's like PHP designed by someone who learned on Visual Basic for Applications

ANYWAY I decided to not just complain about it, and archived it:

<https://t.co/IKxjNQMZMK>

that's the script for this video:

<https://t.co/GPvTp17RDR>

naturally I would comment and add a link to it, but of course I can't post links in the comments.

anyway, autohotkey is great.

have you ever wanted to get a list of drives on a system?

That's simple!

DriveGet, drives, List

now you have a variable named "drives" which is a string containing a list like "ACDEFP"

that's how programming languages work, right?

assigning variables

small brain: `a=1` (fortran, C, descendants of C)

big brain: `a := 1` (ALGOL, Pascal)

cosmic brain: `LET A=1` (early BASIC)

multiverse brain: `a <- 1` (F#, OCaml)

brane cosmology bulk brain: there is no assignment, functions output to one of their arguments (AHK)

fun fact: AHK actually also uses `=` and `:=`

it just doesn't use them for functions.

it's got `"=`" which is called legacy... assignment? they don't actually call it assignment.
legacy variable storage, let's say.

the fun thing is that `=` only takes integers, unquoted strings, and other variables.
so:

```
a=1
b=FOOBAR
c=%b%
```

but if you do:
`a=1+2`

that's either an error, or you'll get a string containing "1+2".
I'm not sure which. I don't really want to find out

whereas if you do:
`a := 1+2`

a will be set to "3"

BTW these differences in `=` and `:=` mean that if you convert that above example from `=` to `:=`, you get:

```
a=1
b="FOOBAR"
c=b
```

in computer science, we call this "consistency"

this is the fundamental problem with AutoHotKey as a language:

it started as a bad batch scripting language and got a little powerful with some visual basic ideas, and then it got "fixed" into being a more usable language, but THEY LEFT THE OLD SYNTAX VALID FOR COMPATIBILITY

it's like having a language that uses Go's syntax but you can always just drop down to

```
10 PRINT "HELLO WORLD"
```

```
20 GOTO 10
```

in the middle of a method

autohotkey is a language that could have been fixed, and they tried to fix it, but they tried to fix it by nailing the "fixed" version onto the side of the broken version

I guess it's fortunate then that microsoft is trying so hard to murder autohotkey.

once it stops working there won't be any more language evolution

maybe I should port autohotkey to an arduino keyboard+mouse.

or maybe I shouldn't torture tiny computers